Percolation within Percolation Simulation

Kit Tyabandha, Ph.D.

Particles suspended in a fluid generally have their diameter d_p much smaller than the pore diameter d_v of a filtering membrane. Yet these small particles can cause blockages of the membranes due to attrition among themselves. Since in this case $d_p \ll d_v$, blockages due to blocking or blinding of the individual particles (*cf* Jackson, 1994; Jafferali, 1995) become out of question.

Having investigated both the percolations of networks and continuum, I suggest that the blockage of these smaller particles in membranes is due to a double percolation phenomena, one the percolation of the suspension continuum, the other the percolation of the centroidal Voronoi network. As a reminder of a centroidal Voronoi network, it is a Voronoi tessellation on generator points which are the centroids of a Voronoi network which either is generated from Poisson point generators or is another centroidal Voronoi network.

Because percolation is a study of the behaviour of two phases, and because in general $p_c \neq 1/2$, there are not only two but no less than three states or regions of behaviour to consider in each percolational investigation (*cf* Tiyapan, 1997; also in Tiyapan, 2003). When $p_c < 0.5$ these three regions are $p < p_c$, $p_c \leq p \leq (1 - p_c)$ and $p > (1 - p_c)$, and when $p_c > 0.5$ they are $p < (1 - p_c)$, $(1 - p_c) \leq p \leq p_c$ and $p > p_c$. The case where $p_c = 0.5$ is assumed to be very rare in nature, and so can be neglected in the present study.

When a suspension becomes so concentrated that the average interparticle distance has become such that the attrition due to van der Waals force is prominent, it will solidify into a moisted bed of particles. According to the percolation theory, we may define the point where this spontaneous solidification occurs to be that point where there is a single cluster, under a mutual van der Waals force, which traverses the whole continuum in a certain well-conditioned direction, that is a direction which may represent the diameter of the network. Furthermore, let us call a critical concentration ρ_c the minimum concentration at which this infinite cluster appears.

Then we have for our suspension a continuum percolation with three regions of behaviour similar to those we have found in the case of network percolation. Furthermore we map the space of ρ_c on to that of $0 \leq p_c \leq 1$, where $p_c = 0$ means there are no particles suspended in the fluid, in other word $\rho = 0$, and $p_c = 1$ is where the suspended particles form a bed in the closest packed structure, that is $\rho = \rho_{\max}$. We also assume, without the loss of generality, that $p_c > 0.5$. Then we have the following as the three regions, $p < (1 - p_c)$, $(1 - p_c) \leq p \leq p_c$ and $p > p_c$.

We are interested in neither the cases $p < (1 - p_c)$ nor $p > p_c$, since the former implies too dilute a concentration for the attrition due to the van der Waals force to cause an infinite cluster, while the latter means that the suspension is so concentrated that they solidify instantaneously, simultaneously in all pores.

When $(1 - p_c) \leq p \leq p_c$, all pores has an equal probability of being solidified, and so the per cent total solidified cells now depends on the topology of the network, in this case Voronoi, and the probability where the critical phase change occurs becomes the critical probability of the network.

In the case of dead-end filtration the flow is in one direction, therefore the critical flux reduction which is the result of percolation of the network occurs at the point where the cross section of the network, not the network itself, percolates. This is because such percolation in the cross section in the plane perpendicular to the flow direction will cause a bottle neck in the flow and therefore determines the flux. This phenomenon is summarised in Figure 1.

Next we must define the critical probability of the overall system. Since the system involves two kinds of probability, that is continuum and network, and assuming p_{c_0} and p_{c_2} are two independent probabilities, then we have the overall probability is

$$p_c = p_0 p_{c_2}, \tag{1}$$

where $(1 - p_{c_0}) \leq p_0 \leq p_{c_0}$.

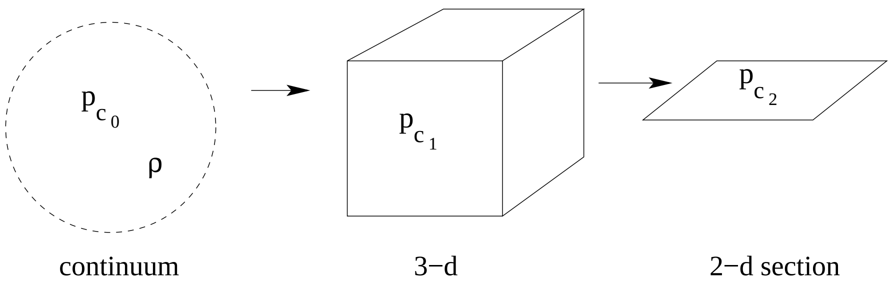


Figure 1 *Percolation within percolation;* $0 \le \rho_0 \le \rho_{\max}$, $0 \le p_0 \le 1$, $p_c = p_{c_2} \cdot p_0$.

If our membrane is homogeneous, the reduction in the area perpendicular to the flow becomes $\Delta A = A_v p_{c_2}$, where $V_v/V_t = A_v/A_t$, V_v and V_t are respectively the void volume and the total volume of the membrane, and similarly for the areas A_v and A_t.

The first part, suspended particles

Because of the complex nature of the problem, there is no single algorithm but rather there is an algorithm for each job. The first task is to study the percolation of the spherical particles under the van der Waals force.

Continuing from the development in Tiyapan (2004), particles are spherical in shape with $r_p = 5\ \mu$m and the capturing radius is $r_c = 167 \times 1.2 \approx 200\ \mu$m. First we shall study particles within a cubic box

of side length 2 mm. Particles start as a suspension with no obvious velocity. They stick together and to the walls.

When particles come together, they form a porous globule, having the densest packing density of the hexagonal or cubic close packing. When this happens, we discard the individual particles and consider instead the globular cluster which they formed. The cluster is porous, so its new radius is $r = (\pi/3\sqrt{2})(3\sum v_i/4\pi)^{1/3} = (\pi/3\sqrt{2})(3nv/4\pi)^{1/3} = \pi^{2/3}(nv)^{1/3}/(3^{2/3}2^{7/6})$.

But we do not know the rules by which these particles stick themselves together, whether they form a closest-packed globule or some other shapes. It is quite certain that whatever shape they are after, they may not retain it for long because there is a limited space within each pore that will put constraints on the way they grow. We shall call the growth of clusters into globules mentioned above *globular formation*.

Other clustering mechanisms possibly include what we shall call the *tetrahedra formation*. By this I mean that each one of our spherical particles attaches itself to three other particles, forming a tetrahedron whose side lengths are two times their radius. The next free particle may fit into any of the available attachment sites, *i.e.* the free triangular faces of an existing cluster. We may suppose that it always choose the closest one among such sites if this is available; if not, then the next closest one and so on.

These are only two among all the possibilities, namely the globular and tetrahedra formations. There could well be others, as well as a mixture of them. On the other hand, each material of which the particles are made may decide the particular cluster shape it prefers. Detailed analyses in thermodynamics and quantum mechanics are needed if we were to understand this cluster formation in continua under spatial constraints. Neither of these is within the scope and time constraint of the present work, though both of them merit a detailed investigation which I plan to carry out in the future.

Another problem arises when we come to consider percolation of our membrane. We may, for instance, say that it percolates when its structure in three dimensions percolates, or we may say that it percolates if there exists a cross section perpendicular to the flow which percolates in two dimensions. Since percolation of a cross section implies percolation of the structure but not vice versa, these two definitions of percolation due to suspension in membranes are not the same.

Choosing the percolation of sections as a criterion implies that we consider the superficial velocity of the flow whereas choosing the percolation in three dimensions as the criterion means that we focus on its interstitial velocity instead.

The algorithms for the study of percolation by tiny particles due to attrition in membranes which proposed here are Algorithm's 1 and 2. Algorithm 1 prepares the structure while 2 does the percolation simulation. Here both VT and $\mathcal{V}$ means the Voronoi tessellation. The appeal factor is the probability that a particle will choose to leave a cell via a certain bond. It is weight by the gradient of each bond, and is calculated over all bonds going in the downward direction from the cell. Transfer grids are square grids which help map the continuous plane at the top layer to bonds connected to it, that is to say, it maps a continuous Euclidean plane into discrete grids and from there on to bonds. In other words, $E^2 \to D^2 \to \{b\}$.

Algorithm 1 *Percolation by tiny particles due to attrition in membranes.*

generate a Voronoi tessellation in three dimensions;

transform the VT into a centroid VT;

find the cross section of its top layer;

$C \leftarrow C^2(\mathcal{V})$;

find transfer grids of C;

for every cell in VT **do**

  find the maximum chamber capacity of its cell;

  find appeal factors for all its bonds;

endfor □

Let the gradient of each bond be represented by an angle α that it makes with the horizontal plane. Then the gradient can be calculated from the coordinates of the two end points of each bond, providing that $z_2 > z_1$, from $\alpha = \tan^{-1}(z_{12}/((\Delta x)^2 + (\Delta y)^2)^{1/2})$, where the slope

is downwards from p_2 to p_1, and as usual $z_{12} = z_2 - z_1$. Algorithm 2 describes the membrane percolation simulation proposed.

Algorithm 2 *Percolation by tiny particles due to attrition in membranes, percolation simulation.*

for each time step **do**
  for all arriving particles **do**
    find their random arrival position;
    round these positions to the precision of the grids;
    map positions on to bond numbers, using the grids;
  endfor
  for all particles **do**
    update distance travelled;
    update chamber crowding;
    find percolation of blocked chambers;
    if chambers percolate **then**
      terminate the simulation;
    endif
  endfor
endfor □

Here we concentrate on the interstitial flow velocity, therefore the percolation is supposed to occur when the chambers percolate in three dimensions. Coincidentally, this also makes the calculation easier. If

we were to choose the percolation of sections as the deciding factor for percolation of the membrane, for instance, we would have needed to consider approximately $2n$ sections in total, where n is the number of chambers. With an equal probability for success for all the homogeneous sections, this would still leave us on average n sections to consider before we know that a sample percolates, if it does, though we would still need to test all the $2n$ sections in cases where it does not. This number $2n$ arises from the fact that to completely cover all combinations of grouping cells into sections we need to consider for each cell two sections for each existing cell, one touching its top while the other touches its bottom.

Sphere packing is a rich field of its own, within which the packing density is generally referred to as η, an efficiency, instead of the usual density symbol ρ. The rigid packings of spheres vary in density from the lowest in loose packing where $\eta \approx 0.06$ to the highest, which is shared by the cubic and the hexagonal closest packings, $\eta \approx 0.74$. A rigid packing is a packing in which all spheres touch at least four others, and the points by which each sphere touches its neighbours can neither be all in the same hemisphere nor all on an equator, *i.e.* a greatest circular section. So we can now limit the value of ρ that we shall use to be in accord with $0.06 < \eta < 0.74$. In this early stage we shall not use a Monte Carlo study to find the probable ρ, *i.e.* η, but will approximate it to be some value within the range mentioned. Since the most familiar sphere packing in human history must be that by which oranges are stacked at markets, especially open markets like the one at Bolton, which gives the efficiency of packing $\eta \approx 0.74$, we shall assume that this is the way the clusters arrange themselves.

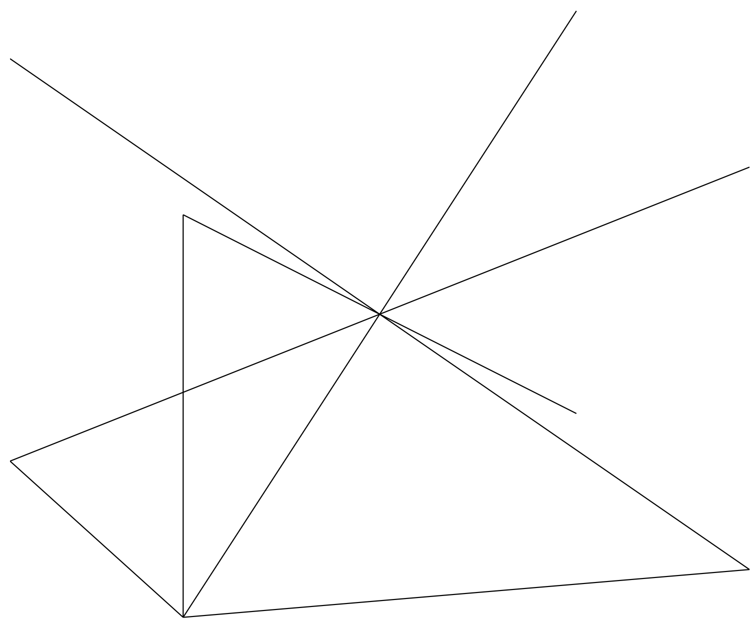
Notice also that piling oranges in a neat tetrahedral shape on a table and a packing them into a rectangular box both produce the same crystal structure, that is the face-centred lattice, the difference being only in their habits.

For all intents and purposes the percolation probability of spheres under the influence of the van der Waals force must be the same as p_c of a face-centred lattice. This is because the biggest cluster of both cases will have the same structure and their orientation will determine the orientation of the structure. We can do away with the orientation of other minor clusters precisely because they are much smaller, which justifies our grossing over their individual shapes and only concern ourselves about their statistics, that is to say, their number. I think that stacking oranges into a box is cubic close packing while a pile of oranges is hexagonal close packing, but this needs to be checked.

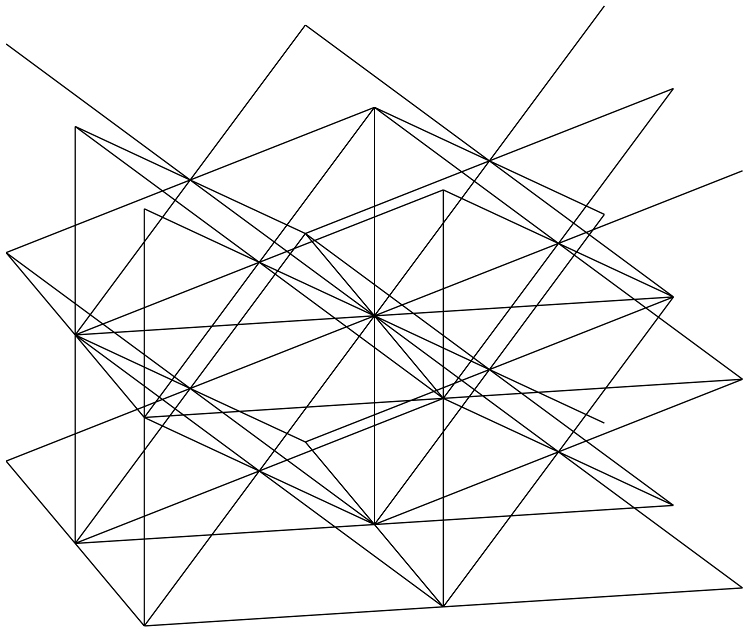
To find p_c of the close-packed cluster of spheres, one needs a programme similar to one of the those mentioned at the end, but which would do the job for three dimensions instead of two. For this purpose, the programme for 2-d tilings has been developed further to deal with regular lattices in three dimensions. At first I thought that there should be some other way to do this instead of having to develop another programme for a general lattice in three dimensions, since this is already the last week of the project and time is running out. But in the end I found it better to spend some time to systematically develop a programme for general cases than to opt for some adhoc approaches. As a result, a programme that creates regular lattices in three dimensions for the purpose of percolation study has been written and is listed at the end.

As the 2-d programme does for all 2-d regular lattices, this new programme can deal with all possible lattices in three dimensions. The difficulty is, however, in the meticulous nature of identifying all the vertices and links in each unit cell. In this respect, the cubic close packing is much simpler to do than the hexagonal close packing. Therefore we shall only do the first one while leaving out the second, which ideally could be used for the purpose of comparison.

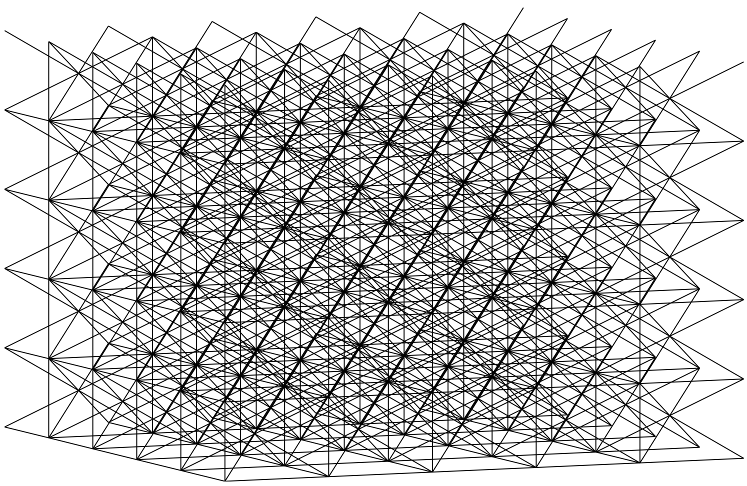
Figure 2 shows the lattice generated by the programme and one which is used for finding the percolation threshold.



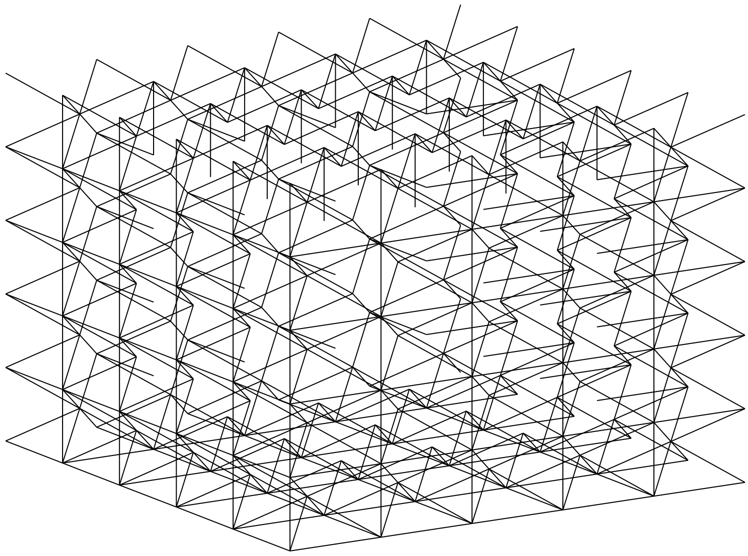
(a)



(b)



(c)



(d)

Figure 2 *Percolation of the cubic close-packed lattice;* (a) *a unit cell,* (b) *eight unit cells, one from each of the eight groups,* (c) *network of size* $5 \times 5 \times 5$ *unit cells, which is used in a simulation, and* (d) *the same network with only boundary edges drawn to make it easier to look at.*

At present the programme only finds p_v, p_e, x_v and x_e, not p_c, p_b, x_c and x_b. There may be altogether three types of cell and bond pairs, in comparison with two in the 2-d case, depending on whether the number of shared vertices required be 3, 2, or 1. For our purpose in the study of filtration, we only need to know p_v, which, when generated from a $5 \times 5 \times 5$ network as shown in Figure 2 (c), turns out to be 0.25. All the results from simulations are $p_v = 0.2501 \pm 0.0400$, $x_v = 8.0645$, $p_e = 0.1320 \pm 0.0209$, $x_e = 17.1709$, while $n_v = 341$ and $n_e = 1{,}375$.

This means that when the space will be blocked, *i.e.* percolates, when it is filled up to one quarter of its volume by suspended particles in the form of clusters of the highest packing density. Because the cubic close-packed spheres fill 0.74 of the space, this ratio translates into the real volume ratio of $0.74 \times 0.25 = 0.185$, that is 18.5 per cent by volume. If the fluid in our system is water, then $\rho = 1{,}000\ \mathrm{kg \cdot m^{-3}}$ and the percentage by volume above is equivalent to a density of the suspended particles of $555\ \mathrm{kg \cdot m^{-3}}$. Notice also in our simulation that the cluster shapes need not be convex.

In the light of the symmetry between particles and space, in other words between particles and anti-particles, which give rise to a symmetry and the three types operational regions that I originally proposed in a study of traffic congestion (*cf* Tiyapan, 2004), the operational space of our filter may fall into three distinct regions when it is subjected to very small particles suspended in a fluid.

If we specify by ρ_v the ratio of the volume occupied by all the clusters to the total volume, and ρ the density in weight per volume, and if the three regions of operation are labelled I, II and III, then in the case of I, $0 \leq \rho_v < 0.25$, while for II, $0.25 \leq \rho_v < 0.75$ and for III, $0.75 \leq \rho_v \leq 1$. In other words, for I, II and III, we have respectively $0 \leq \rho < 555$, $555 \leq \rho < 1,665$ and $1,665 \leq \rho \leq 2,220$, where the unit of ρ is kilogram per cubic metre. Generalising this, the regions I, II and III correspond respectively to $0 \leq \rho < \rho_{c_1} = \rho_1$, $\rho_1 \leq \rho < \rho_{c_2} = \rho_2$ and $\rho_2 \leq \rho < \rho_c$, where ρ_c is determined by a physical constraint, namely the packing efficiency mentioned.

Qualitatively speaking, these three regions may correspond to the operational-, blocked and non-operational regions. If our system also contains other particles which are larger than the pores, then in Region I filters will operate normally until the effects of blocking or blinding by the large particles become prominent, as has been studied in literature (*cf* Jackson, 1994; Jafferali, 1995). In this case fouling of the filter is caused exclusively by the blocking or blinding of these larger particles, which result in the formation of cake, and unless $\rho = 0$ there will be some blockages of internal pores due to the blockage caused by small suspended particles forming cluster. This latter type of blockage, which is of our concern, is to be expected due to various reasons. Cluster formation may be caused by nonhomogeneity in the concentration of the suspension which raises the concentration in some region such that it exceeds ρ_1.

Additional reduction in the flux is to be expected from two reasons. Firstly blocking clusters may block some of the pores. And secondly, the suspended particles together with free clusters, *i.e.* those which are smaller than they could block pores, displace the volume of the liquid surrounding them and thereby reduce the flux. The first one of these will produce an effect similar to blinding described in literature (*cf* Jackson, 1994; Jafferali, 1995, *ibid.*), where no backflushing may recover the filters to their virginal state. On the other hand, the majority of those particles and clusters in the second scenario is expected to be easily removed when backflushed. Among these latter there could yet be some which adhere themselves to the walls, whose fixation defies backflush-

ing. But these last ones are expected to be small in number, and thus can be neglected, because we shall assume that the combination of ρ and channelling results in the probability that pores are blocked being very close to either zero or one.

The channelling effect, or the occurrence of rivulets by some other authors, is the formation of preferred paths through a porous media through which liquid and the solids it contains pass. It is not yet clear what these rivulets would do to our system. If all the channels channel equally both the liquid and the solids, then the blockage along these path may be expected to rise above the average value of the whole structure if only because this becomes more probable statistically. But if there exist some channels which prefer channelling liquid to particles or vice versa, then the effect they produce will vary and become complicated. For example, channels which like to channel particles are more likely to find themselves blocked in the end by those particles which pass through them. On the other hand those channels which channel liquid better than solids will be less prone to blocking on average, but will leave other pores around them with the excess particles, and these latter will necessarily become blocked more often than usual. But, for our purpose here, we shall assume that it is solid particles that are being channelled. This should raise the possibility of blocking in some of the pores by certain amount. Channelling in general needs further investigation which will not be covered here.

Before going on to the next step of our study we should briefly look at the core idea that makes the programme used. There are eight types of unit blocks here, compared with the four types in the case of the programme listed at the end. These correspond to the area drawn and labelled in Figure 3.

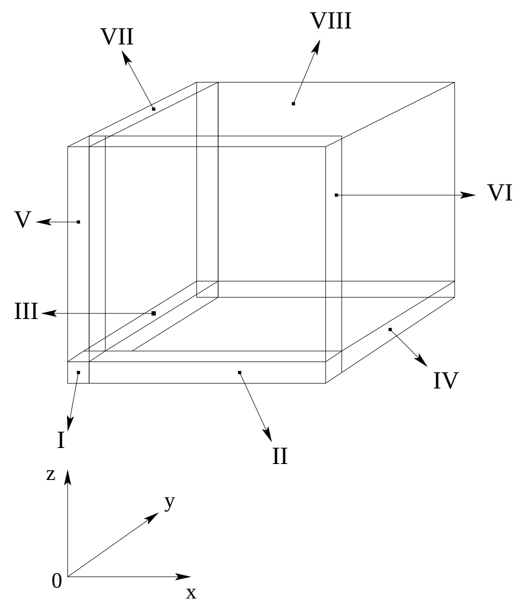


Figure 3 *The eight areas defined by the eight types of unit blocks*

Figure 3 shows the eight areas defined by the eight types of unit blocks they contain. Area's from I to IV correspond to those previously defined for the 2-d programme. Although unit blocks in the various areas works differently, that is to say, they adopts different set of vertices from different sources, and create different edges, all of them follow the same four rules. These four rules in the mnemonic not cryptic forms which I use are, *take vertices from behind*, *make front vertices*, *draw edges behind* and *draw no front edges*. With these rules in mind, both programmes should become self-illuminating to such extent that no further explanation is needed. This set of rules does two things, namely organise vertices and then link them with bonds. The unit vectors in the three directions being orthogonal means that we will have a nice and square end product suitable for a percolation study.:

In fact it is wrong to say that all four rules work differently for each unit group. Only the two on vertices, *viz.* the first two, are different. The rest, *viz.* the last two which concern edges, are the same for all basic units. These are the only two crucial tasks with the discovery of

which the writing of both programme becomes worthwhile.

The input data needs only contain details regarding units in Area II, III and V. Area I, being at the origin, is trivial, or should one rather say unique. Area IV can be derived from Area's II and III. Like wise VI is derived from II and V, and VII from III and V. Finally Area VIII turns out to be nothing but II, III and V combined.

So far we have only mentioned the situation where $0 \leq \rho < \rho_1$. In the second case, where $\rho_1 \leq \rho < \rho_2$, many more pores are blocked from small particles than in the first case. The concentration is already beyond the first critical point. But while the second critical point is still not reached, there would still be an infinite cluster of space – in this case the liquid – surrounding the particles. In other words the space still percolates. The presence of this infinite cluster, or continuum of the medium, means that the filter can still be in operation until it should be blocked or blinded by larger particles which individually can physically block the pores as found in existing literature earlier mentioned.

The third and last case, where $\rho_2 \leq \rho \leq \rho_c$, represents the extreme which can be easily comprehended. Here the solid particles occupy more space than the liquid does, as a result of which the combination is no longer a suspension but a slurry. Clay material produced by this slurry would block most of the pores within the structure and make filtration impossible. Backflushing will not be effective on filters which have undergone such fate.

The second part, flow through the cells

Next we will investigate briefly the effect that channelling may have on the value of p_c. There are three cases considered here. The programme for this purpose is listed in Tiyapan (2004). The programme creates a centroidal Voronoi tessellation in three dimensions. The first set of simulations works on a normal case of cell percolation, similar to that used by Tiyapan (2004) but here the system is a centroidal Voronoi, which implies a constraint on cell sizes and distribution. The function `perd` in it is obtained by inputting the `Blocked` variable instead of generating it internally.

This section is written along with my developing the programmes, so the contents should be easier to follow than in other sections. This is not to say that in those other sections I had not kept records of things discovered. Every Ph.D. student starts off doing his project knowing

that he should write along as he goes, and plans to nothing but that. But the truth is that even though we always write, but the way we write develops with our experience. Also, with the increase in the understanding of our problem, we no doubt would be able to write a better description of what we do and how we do it. This is unavoidable, and it is probably the reason why we should keep on working.

So much for an aside. From one hundred generators originally, Voronoi operator is applied twice. After the rims has been trimmed there are 280 centroidal Voronoi cells remaining, and this is the value of n_c. For x_c the value is 10.7929, while p_c from 2×5 simulations is 0.2314 ± 0.0602.

Next investigate the effect of channelling by assuming that the steepest gradient of all the bonds arriving at a cell decides how quick it percolates. Here cells are sorted according to their steepest gradient of incoming bonds. Working on the same network as previously, if the percolating order is such that the steeper the quicker, then $p_c = 0.2107$. But if on the other hand steeper incoming bond means slower percolation, then $p_c = 0.1429$. The critical probability is constant in this case since the order of percolation is predetermined by the orientation of bonds with respect to cells.

If instead of looking at only a single bond we take the signed summation of bonds entering and leaving a cell, then $p_c = 0.1321$ when the criterion is $\min(\sum b_i - \sum b_o)$, and $p_c = 0.1393$ when it is $\max(\sum b_i - \sum b_o)$, where b_i and b_o are respectively the incoming and outgoing bonds.

Our studies up to now tell us that if the suspension is homogeneous and there is a rivuleting effect, then the location of the blockages made by clusters of suspended particles among all pores of the structure is predetermined. This is in contrast with the blinding and blocking of large particles, where such location is random. In fact, even for these latter large particles, the location can only be random when the particles have a variety of sizes. It can never be wholly random, however, if this is not the case, since in the former case the randomness is introduced by the distribution of sizes which is random, but in the latter the blinding or blocking will be determined by the size of pore openings which is fixed by the geometry of each network. The randomness then can only be in the order not location of blockings.

The next study is a combination between continuum and network percolations. Here suspended particles are grouped into quanta, each channelling through a path or rivulet of interstitial distance with some

interstitial velocity.

At the top, the layer of the structure where the incoming particles arrive, is a cross section all the cells of which are gridded to provide means to determine the path at the beginning of each quantum. The partitions in this layer is conveniently found by cutting some faces of the convex hull of each cell in the top layer by the plane $z = 0.9z_{\max}$, where $z_{\max}$ is the maximum z-coordinate of all the cells under consideration.

Given coordinates of two points, (x_1, y_1, z_1) and (x_2, y_2, z_2), and a plane equation $z - a = 0$, we may think of the plane equation as being one coordinate given, $z = a$, and find the coordinates of intersection between a line passing through the two points and the plane from the parametric equations for the line. Parametric equations are in fact interpolation done on each of the coordinates. In this case, which is useful when finding the intersection between an edge of a triangle and a plane perpendicular to some coordinate axis, the parametric equations are $x = x_1 + x_{12}t$, $y = y_1 + y_{12}t$ and $z = z_1 + z_{12}t$. From the plane equation $z = a$, therefore

$$t = (a - z_1)/z_{12} = (a - z_1)/(z_2 - z_1)$$

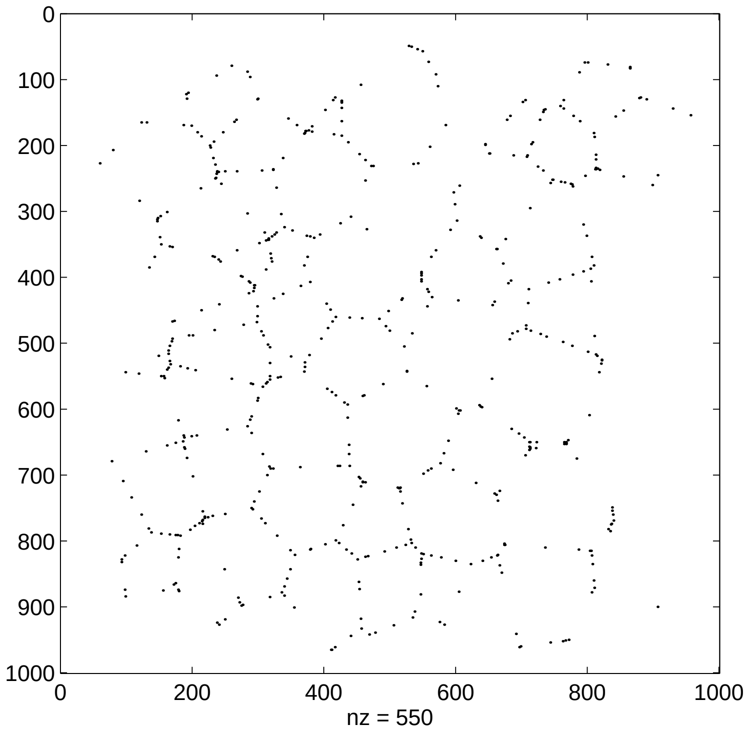


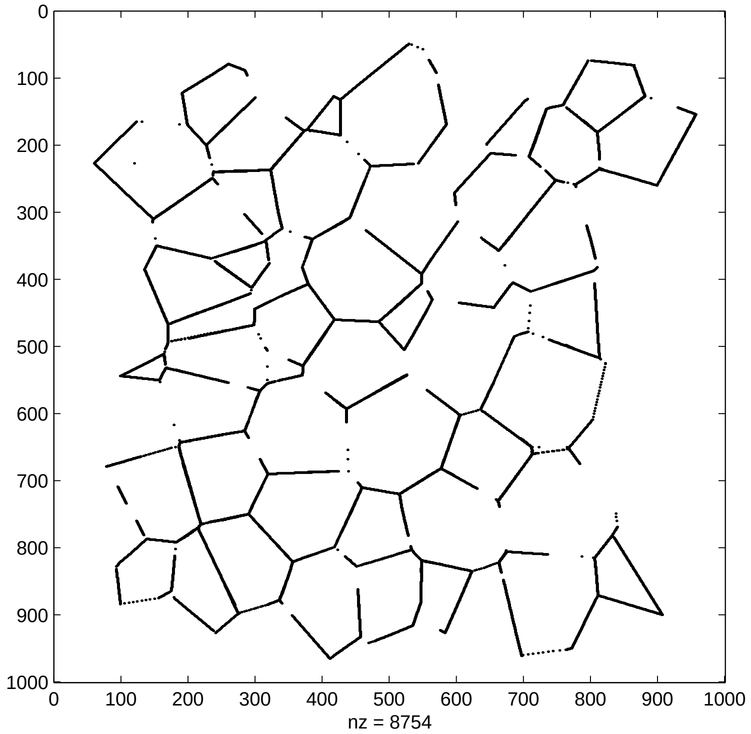
Figure 4 *Intersection of convex hull faces and the horizontal plane*

The programme being written finds the intersection of cells with the horizontal plane by finding the intersection of the faces of its convex hull with the same. Since every face of the convex hull is a triangle, the programme essentially finds intersection between edges of these triangles and the horizontal plane. The result obtained from an intermediate state during the course of development of the programme is shown in Figure 4. The partitions look incomplete because the picture is taken as a test while developing the programme as mentioned. A picture with the same degree of incompleteness as this one is not to be obtainable from the completed programme.

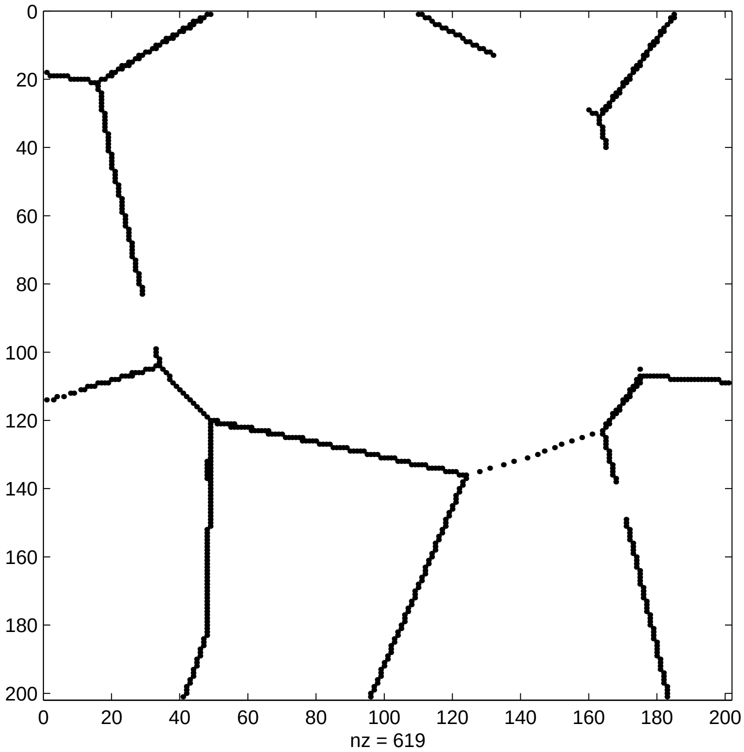
For each face of the convex hull that intersects the plane, there will be two points of intersection arising from the two edges of the triangle intersecting it. Let (x_1, y_1) and (x_2, y_2) represent these two points. Then we may scan up in the y direction finding $x = x_1 + (y - y_1)x_{12}/y_{12}$ for each y along the way, and then scan in the x direction, this time finding instead $y = y_1 + (x - x_1)y_{12}/x_{12}$. Afterwards we could fill in the space

by scanning along the x direction for all y positions.

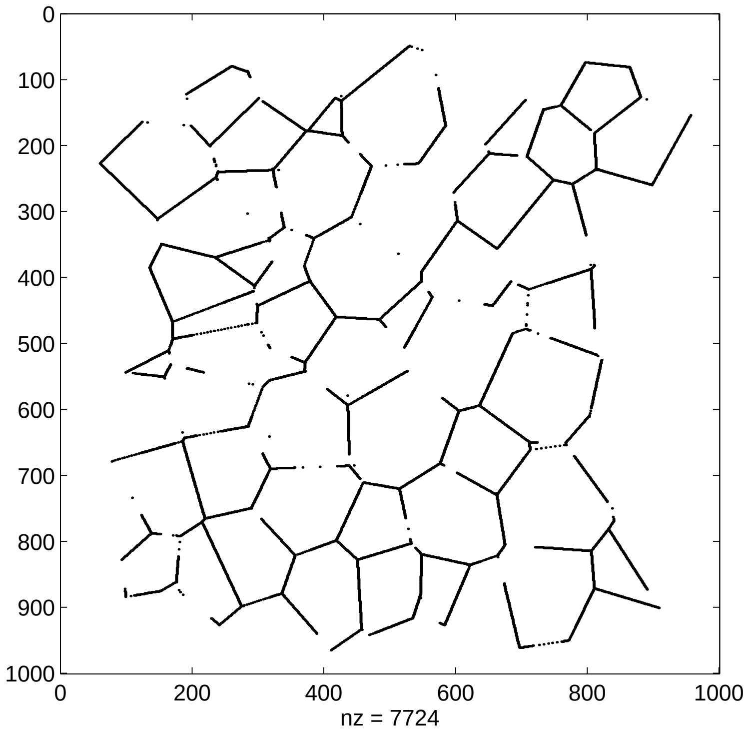
Figure 5 shows the progress of my programming the codes, step by step, trying to close all the partitions such that no gaps remain. Two problems have been discovered, namely those of rounding and precision. Before the correction the result looks like Figure 5 (a) and (b), and after rounding problem corrected Figure 5 (c).



(a)


0
20
40
60
80
100
120
140
160
180
200
0 20 40 60 80 100 120 140 160 180 200
nz = 619

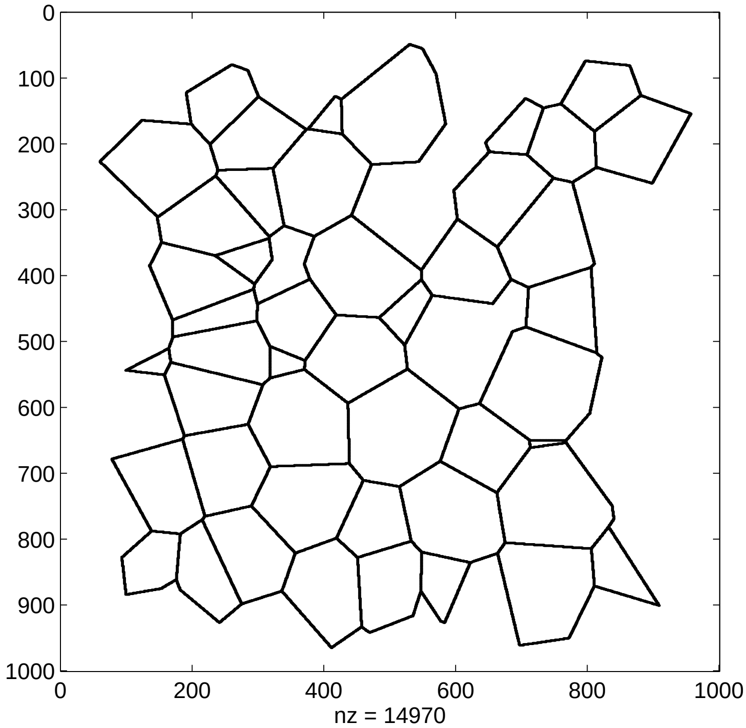
(b)



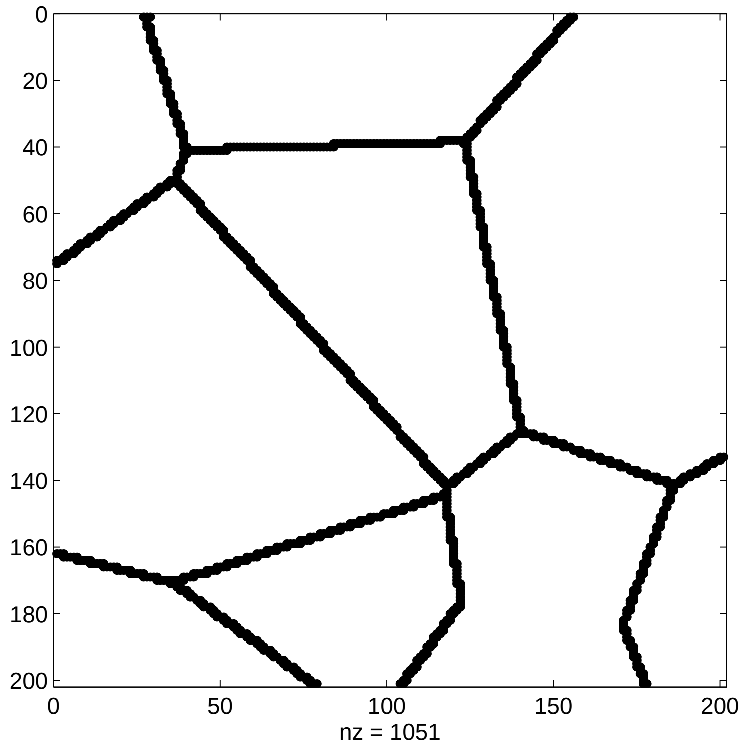
(c)

Figure 5 *Correcting the effect of rounding, in other word discretisation.* (a) *The gaps resulted from rounding or discretisation,* (b) *a closed-up view of (a) and* (c) *partial remedy where roundings have been solved but with the degree of precision not yet raised.*

After having corrected the problem regarding precision, by increasing the number of steps when calculating x or y, the result still misses several walls, the cause of which is still unknown at present. This is shown in Figure 6.


0
100
200
300
400
500
600
700
800
900
1000
0
200
400
600
800
1000
nz = 14970

(a)



(b)

Figure 6 *After having increased the precision to ten times the previous value,* (a) *and a closed-up view,* (b).

It seemed at first that there might be some triangular faces missing from the surface of the convex hull, which would have accounted for the missing boundary in the section. But after having tested the minimum number that an edge of each hull appears as the edges of all its triangular faces, and see that it's value is correctly two, this becomes out of question.

The figures, *viz.* Figure 4, 5 and 6, are produced from the `spy` command in Matlab, as a result of which the x-axis runs downwards while the y-axis runs to the right. This command looks at a matrix from above as we look at a map. In the present case our matrix is a full-, not sparse matrix. The number written at the bottom is the number of all its nonzero components, which is less than the number of times that we calculate them since we need to calculate some of the points more

than once in order to increase the precision to eliminate gaps in other places.

The command `spy` is used more often with sparse matrices since these are often too large to list, and listing their members in pairs makes it difficult to visualise. As an example, Figure 7 (a) is what we get when we `spy` our neighbour matrix `necc`, while in Figure 7 (b) are all the neighbours that the 100$^{\text{th}}$ cell has. Cells which have few neighbours generally live along the border. For example the 110$^{\text{th}}$ cell has only three neighbours, and it is located not far from the lower x limit.

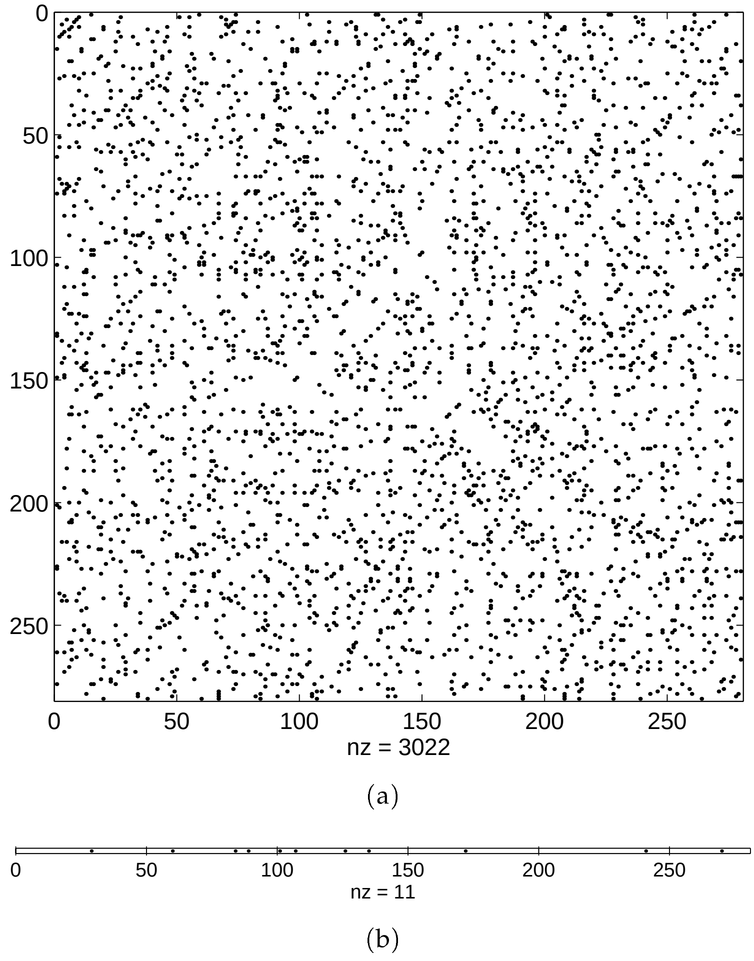


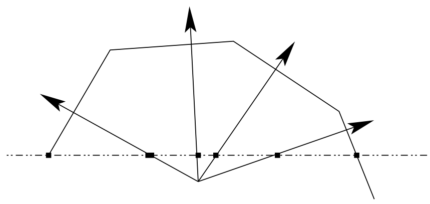
Figure 7 *The result when we* `spy` *our neighbour matrix. Here* (a) *shows the neighbours of every cells while* (b) *only shows those of the* 100$^{\text{th}}$ *cell.*

At this stage the neighbour matrix of the programme is double-checked, and find that it includes poorly defined neighbours, that is those which only have one vertex in common. Therefore the programme is first altered to make it only look for neighbours who share at least two vertices. But this has shown no noticeable changes in the results that we have so far.

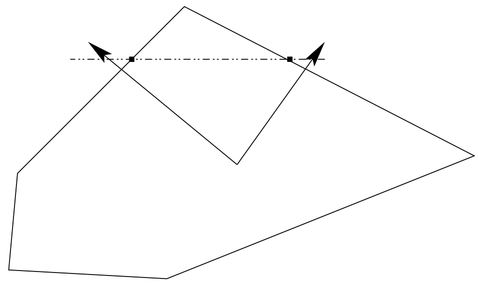
The next step is to colour the cells. Like a child painting a picture, there are so many ways one can paint or label the tiles of a tessellation. For example one could draw a vertical line first and then branch out either horizontally or diagonally. On the other hand, one could also draw the diagonals first. Yet another way is to expand radially, spiralling outwards. With parallel computing we could also divide the area into domains, paint each domain, and then merge the resulted areas together.

But here we opt for painting the cells by scanning horizontally, moving upwards in layers. Once past a wall, the programme moves on until the next wall is reached while gathering all the grids that are between the two walls into one group. It then colour the whole group by a colour picked up from one layer below it. If no colours exist, then it creates a new colour which in turn gradually propagates upwards this way until the upper wall is reached.

(a)



(b)



(c)

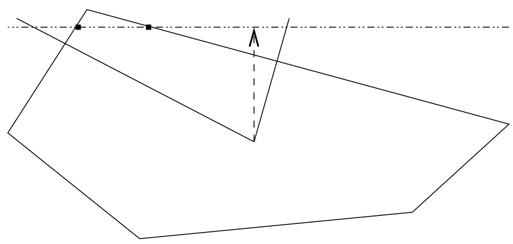


Figure 8 *Sections of bonds and cells*

The intersection of bonds will not always coincide with the intersection of the cell, neither is the projection of a cell perpendicular to the plane the sectional plane of the cell. This is because of the three possible situations shown in Figure 8. In Figure 8 (a) the cell section contains several points, while both (b) and (c) contain none.

And here sadly the time runs out, so I will suffice myself to describing what I see should be done next. Up to now we have a three-dimensional network and its top section. We also have the list of all its

bonds, which contains the connections and the draining angles sorted in a descending order.

Next we should find a mapping from each cell section to the corresponding nuclei. Then the times it takes to traverse each bond must be calculated. This time for each bond is then divided by half, one belonging to each of the two cells connected by the bond.

When it comes to bombarding the filter with our small particles, we can not keep track of millions of particles and therefore we should quantise them into units. These units or quanta can then be treated as individual particles. When a quantum enters a cell, it is assigned t, the time to reach the nucleus. Later time sees this t decreases in steps until it finally reaches its destination, the nucleus. Once there, it is assigned the next bond to go along, taking into account what bonds are available at the time and their comparative probabilities, which in turn depend on their gradient as mentioned. When this is decided, it is given t_1, the time it would take to reach the border that lies at mid point of the bond.

This goes on forever, apart from that at each time step we look to see whether the blockages in our filter has percolated. After updating the list of blocked cells, if we find that percolation has occurred then the simulation would end. Percolation occurs in each cell whenever its concentration has reached a certain value. This value we have found to be 18.5 per cent by volume.

To calculate the flux decrease we find instead the decrease in the superficial area. This is calculated from the total volume of the void subtracted by the volume of all cells that had percolated, and then subtracted by the total volume of solid particles which are suspended inside the network. The area of the cross section is then the volume which remains divided by the thickness of the filter.

It is not a little to have to leave things unfinished after having started it off. But as one New Zealander poet says, 'Alone we are born and die alone, yet see the red-gold cirrus over the snow mountains shines. Upon the up-land road ride easy, stranger. Surrender to the sky your heart of anger.' And with this we go on to the next chapter.

To summarise, we have begun our study from the Voronoi network, and then went on to study continua and then tried to combine them together. I have suggested the idea that a percolation in continua

can be represented by a percolation in a lattices the type of which depends on the attrition mechanism of the particles in that continuum, in other words the way they pack together.

Bibliography

N. M. Jackson. *A mathematical model to simulate the structure and performance of porous media.* PhD Thesis. UMIST, Manchester, 1994.

Riaz Jafferali. *A stochastic model to simulate the structure and performance of microfiltration media and the growth of animal cell cultures.* PhD Thesis. UMIST, Manchester, 1995.

K Tiyapan. Modelling of traffic congestion. submitted to the Journal of Statistical Physics. 4^{th} November 1997 .

K N Tiyapan. *Percolation within percolation and Voronoi Tessellation.* Kittix, Bangkok, 2003

K N Tiyapan. Division of space by Voronoi graphs, percolation within percolation and application to the models of porous membranes. *Ph.D. Thesis.* University of Manchester, 2004